

technical

Supporting Enterprise Networks and Operating Environments

SUPPORT

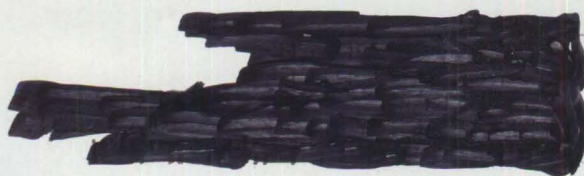
APRIL 1995

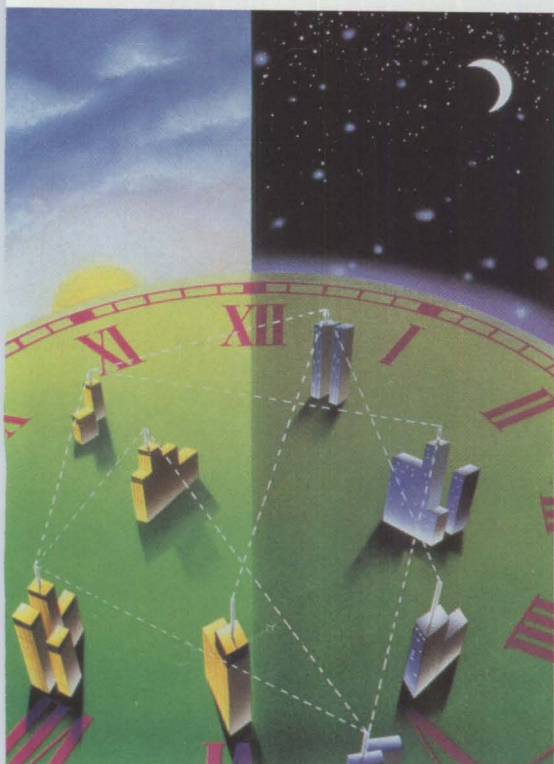
Volume 3, Number 4

**File Transfer on a
Shoestring Budget**

**Cross-Platform Productivity?
The Infrastructure Issue**

**Electronic Software
Distribution Made Easy**





NaSPA Mission Statement:

The mission of NaSPA, Inc., a not-for-profit organization, shall be to serve as the means to enhance the status and promote the advancement of all corporate computing technical professionals; nurture member's technical and managerial knowledge and skills; improve member's professional careers through the sharing and dispersing of technical information; promote the profession as a whole; further the understanding of the profession and foster understanding and respect for individuals within it; develop and improve educational standards; and assist in the continuing development of ethical standards for practitioners in the industry.

The information and articles in this magazine have not been subjected to any formal testing by NaSPA, Inc. or Technical Enterprises, Inc. The implementation, use and/or selection of software, hardware, or procedures presented within this publication and the results obtained from such selection or implementation, is the responsibility of the reader.

Articles and information will be presented as technically correct as possible, to the best knowledge of the author and editors. If the reader intends to make use of any of the information presented in this publication, please verify and test any and all procedures selected. Technical inaccuracies may arise from printing errors, new developments in the industry and/or changes or enhancements to components, either hardware or software.

The opinions expressed by the authors who contribute to **NaSPA Technical Support** are their own and do not necessarily reflect the official policy of NaSPA, Inc. Articles may be submitted by members of NaSPA, Inc. The articles should be within the scope of host-based, distributed platforms, network communications and data base, and should be a subject of interest to the members and based on the author's experience. Please call or write for more information. Upon publication, all letters, stories and articles become the property of NaSPA, Inc. and may be distributed to, and used by, all of its members.

NaSPA, Inc. is a not-for-profit, independent corporation and is not owned in whole or in part by any manufacturer of software or hardware. All corporate computing professionals are welcome to join NaSPA, Inc. Membership rates are \$39.95/year (USA), 49.95/year (Canada) and \$59.95/year (all other countries). \$25.00 of your annual dues is allocated to the publication **NaSPA Technical Support** and is non-deductible therefrom.

NaSPA Technical Support (ISSN 1079-3135) is published monthly by Technical Enterprises Inc., 4811 S. 76th St., Suite 210, Milwaukee, WI 53220-4362. Second-class postage paid at Milwaukee, WI. POSTMASTER: Send address changes to **NaSPA Technical Support**, 4811 S. 76th St., Suite 210, Milwaukee, WI 53220-4362.

All product names mentioned in this publication are the trademarks/registered trademarks of their respective manufacturers.

FEATURES

8 File Transfer on a Shoestring Budget

Visual Basic and Attachmate Extra! together provide an inexpensive solution for automatic mainframe data downloads.

By Bruce Heckler

21 Cross-Platform Productivity? The Infrastructure Issue

Solutions that bypass TSO exist for IS organizations faced with the challenge of leveraging the power of the PC to enable mainframe-to-PC file transfers.

By Russell Shelton



COMMUNICATIONS

42 Networking Multimedia: Part I

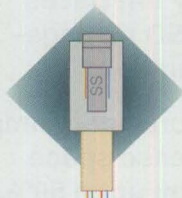
As multimedia technology makes its way into the mainstream, mass distribution of multimedia data over networks is currently at the mercy of a network's transfer rate and bandwidth.

By Guy Yost

48 Communicating With Windows-based Software

With many Windows-based fax and communications products on the market, Delrina Corporation provides a formidable competitor with its Communications Suite 2.1 product.

By Al Shing



CONNECTIVITY

46 NetWare 4.x: A Spy's Perspective — Part I

Will Novell and AT&T provide a competitor to the Internet with NetWare 4.x?

By John E. Johnston



ENTERPRISE

18 Business-Wide Disaster Recovery Planning

As technology and business change, so too must your business's disaster recovery plan to include input from "core business" divisions.

By Leo A. Wrobel

24 Open Systems Standards Soup?

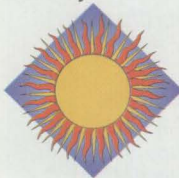
A tongue-in-cheek look at open system standards: application compatibility, competing GUIs and interconnectivity.

By Harold Hauck

27 OS/2, UNIX and Oracle: An Unlikely Combination? Part II: UNIX Shell Scripts

Setting up UNIX shell scripts makes it possible to execute more than one command, be it in an Oracle or OS/2 environment, or any UNIX function.

By Robert Simpson



INSIGHTS

15 CICSplex, Sysplex, Perplexed? Part I

With the 9672 Parallel Transaction Server, IBM has tried to make MVS a financially-competitive alternative to client/server technology, all while offering the benefits MVS users have come to expect: security, reliability and integrity.

By Wayne Bucek

32 A Comparison of CMS Programming Interfaces: Part II

CMS's PIPELINES, similar in concept to UNIX and DOS piping, is a record-oriented processor of data useful for many system and application tools.

By David Kreuter



STRATEGIES

12 MVS Performance Tuning and Capacity Planning: Part I

The need for performance tuning and capacity planning in an MVS environment boils down to a cost vs. benefit situation: customer service and extending the life of the existing mainframe.

By Neil Ervin

36 Electronic Software Distribution Made Easy

Whether your software distribution needs are as simple as a batch file or as complex as a mainframe-based software distribution system, electronic software distribution is flexible enough to meet your disparate needs.

By Eric Allred

COLUMNS

56 Book Review

"Firewalls and Internet Security: Repelling the Wily Hacker"

By Mark Bell

57 MVS Tools & Tricks

VTOC Tidbits — Part I

By Sam Golob

60 VM Toolbox

Mapping CMS Storage

By John D. Kinne

62 VSE Tools & Techniques

A Storage Acquisition Routine

By Mark Hanna

64 Storage Management

UCBs: An Old Control Block Enters the Modern Era

By Steve Pryor

66 NetWare News

NetWare/IP

By Guy Yost

68 OS/2 Insights

Remote LAN Connectivity

By John E. Johnston

70 Security Strategies

Caller ID for Remote

Access Security:

Part I — The Technology

By Eric Allred

71 Open Systems Solutions

Open Systems Desktop Technology

By Harold Hauck

74 Opening Windows

Windows and the 16550 UART

By Al Shing

75 On a Personal Note

Selling Yourself and Taking Risks

By Donald R. Fowler

and Michael K. Sutton

DEPARTMENTS

6 From the Editor

7 NaSPA News

53 BBS Buzz

77 Product Profiles

FROM THE EDITOR



Dear NaSPA member;

You're the Boss!

Feedback, like it or not, is an essential part of our lives. Take for example the feedback you receive from your boss on your day-to-day performance. While these appraisals can be nerve-wracking, in most situations, they provide a valuable gauge of what you are doing right and what can be improved. I'm no different, except I have more than 30,000 bosses. I look for feedback on the quality of what I do on a day-to-day basis: provide insightful, informative, "how-to" articles in *Technical Support* magazine. As one of my bosses, why don't you tell me how I'm doing.

This month, you're the boss! Use the bottom of this letter as your sounding board to tell me what you think about the new, expanded format of *Technical Support* and what you would like to see in upcoming issues. You can even give me an overall grade!

My goal is to provide information on a variety of topics. With so many changes taking place in today's computing environments, your input is instrumental in guiding the direction of *Technical Support* magazine.

Please take a minute to complete the short questionnaire at the bottom of this page and fax it back to my attention at (414) 423-2433. While you're at it, take advantage of the comment lines at the end of the questionnaire. Let me know what you thought of a particular issue or article.

Any time you have a question or comment about *Technical Support*, feel free to contact me. I can be reached at (414) 423-2420 Ext. 123, via NaSCOM ID EDITOR or CompuServe ID 70373,1513.

Sincerely,

Amy Birschbach
Editor, Vice President of Editorial Services

PERFORMANCE APPRAISAL

PLEASE FAX THIS PAGE TO: EDITOR AMY BIRSCHBACH AT (414) 423-2433.

Your Title: _____

Job Responsibilities (include systems you work with): _____

Your Environment (please check only one):

- ☐ Manufacturing ☐ Banking ☐ Government ☐ Healthcare ☐ Services
☐ Other _____

I find the following topics most helpful to do my job (please check all that apply):

- | | |
|---|---|
| <input type="checkbox"/> Security | <input type="checkbox"/> Network Operating Systems |
| <input type="checkbox"/> Network Administration | <input type="checkbox"/> Telephony |
| <input type="checkbox"/> MVS Performance/Tuning | <input type="checkbox"/> Internet |
| <input type="checkbox"/> VM/CMS | <input type="checkbox"/> VSE |
| <input type="checkbox"/> PC/Mainframe Remote Connectivity | <input type="checkbox"/> Multimedia Development |
| <input type="checkbox"/> Programmer Productivity Tools | <input type="checkbox"/> Storage |
| <input type="checkbox"/> Client/Server | <input type="checkbox"/> Mainframe/PC File Transfer |
| <input type="checkbox"/> AS/400 | <input type="checkbox"/> UNIX |
| <input type="checkbox"/> Database Management | <input type="checkbox"/> Debugging |
| <input type="checkbox"/> RS/6000 | <input type="checkbox"/> Other _____ |

In *Technical Support*, I would like to see more (check all that apply):

- | | |
|---|---|
| <input type="checkbox"/> Installation-specific articles | <input type="checkbox"/> Cross-platform articles |
| <input type="checkbox"/> Product profiles | <input type="checkbox"/> Articles on NaSPA, its services and members |
| <input type="checkbox"/> Interviews with industry leaders | <input type="checkbox"/> Interviews with other IS professionals (NaSPA members) |
| <input type="checkbox"/> Product comparisons | <input type="checkbox"/> Other _____ |

My overall grade for *Technical Support* is _____ (Scale: A, B, C, D, F)

Comments: _____

© 1995 Technical Enterprises, Inc.

All Rights Reserved.

Published exclusively for NaSPA, Inc.

BPA audit applied for December 1994.

4811 S. 76th St., Suite 210

Milwaukee, WI 53220-4362

(414) 423-2420 FAX: (414) 423-2433

APRIL 1995 VOLUME 3, NUMBER 4

PUBLISHING

Editor, Vice President of Editorial Services:

Amy B. Birschbach, ext. 123

NaSCOM ID: EDITOR

CompuServe ID: 70373,1513

Internet ID: editor@nascom.com

Assistant Editor:

Matthew Ringlien, ext. 125

NaSCOM ID: EDIT1

Internet ID: edit1@nascom.com

Production Coordinator:

Lisa M. Paulin, ext. 124

Technical Editors:

Eric Allred, Mark Bell,

Craig Collins, Danal Estes, Israel E. Gotay,

Mark Hanna, Howard Hauck,

John E. Johnston, John D. Kinne,

David Kreuter, Leo Langevin, Jim McMaster,

Dwight S. Miller, Steve Pryor,

Stephen L. Samson, Fred Schuff, Al Shing,

Richard B. ViPond, Guy C. Yost

Editorial Assistant:

Debbie Flatow

SALES/MARKETING

Jerry Seefeldt

Vice President of Advertising Sales, ext. 110

Display Advertising, Card Decks, Reprints

NaSCOM ID: MARKET

Internet ID: market@nascom.com

Cal Hart

East Coast Sales

Display Advertising, Card Decks

(908) 495-6660

Steve Cecil

West Coast Sales

Display Advertising, Card Decks

(415) 595-285

Mike Czarnecki

Sales Representative/

List Manager, ext. 105

Mailing List Sales, PC Merchandise Sales

NaSCOM ID: PCSALES

Internet ID: pcsales@nascom.com

Cover Credit: Salem Krieger

OS/2, UNIX and Oracle: An Unlikely Combination? Part II: UNIX Shell Scripts

Setting up UNIX shell scripts makes it possible to execute more than one command, be it in an Oracle or OS/2 environment, or any UNIX function.

This article introduces UNIX shell scripts and provides a number of examples which can be useful in an Oracle environment. Part III and Part IV will expand this approach to managing the Oracle DBMS using scripts written for one of the Oracle utilities, SQL*Plus.

Although this article contains examples specific to an Oracle DBMS, this approach can be used to perform any functions under UNIX. Any UNIX command line utilities or applications can be used in your scripts. The primary objective is to provide a way to execute these scripts from OS/2 as well as UNIX.

Figure 1: OS/2 REXX Command Procedure
u:\cmd\cmdo.cmd

```
/* CMDO.CMD */
/* Execute UNIX command with "oracle" user ID */
parse arg args
"@rexx yourHostname -l oracle -p password" args
exit
```

Figure 2: UNIX C Shell Script /util/bin/chmodtrc

```
cd /u/oracle/rdbms/log
chmod 644 $*
```

Figure 3: OS/2 REXX Command Procedure u:\cmd\bino.cmd

```
/* BINO.CMD */
/* Execute UNIX shell script with "oracle" user ID */
parse arg script parms
file = 'u:\bin\script'
if stream(file,'c','query exists') == '' then do
  say script "is not a valid UNIX script name"
  exit
end /* Do */
tempdir = value("TMP","OS2ENVIRONMENT")
if tempdir \= '' then tempdir = tempdir || '\ '
call value "LAST_BIN_SCRIPT",script,"OS2ENVIRONMENT"
"@echo off"
"echo bino.cmd" script parms ">"tempdir||script".o"
"rexx yourHostname -l oracle -p yourOraclePassword
+ /util/bin/script parms ">"tempdir||script".o"
"@type" tempdir||script".o"
exit
```

Note:

+ indicates a line which is shown as a separate line but should be typed as a continuation of the previous line

EXECUTE UNIX COMMANDS AS "ORACLE"

Part I proposed setting up a "repository" containing OS/2 REXX command procedures, UNIX shell scripts, Oracle scripts and documentation files separated by placing them in different subdirectories such as "/util/cmd", "/util/bin", "/util/dba" and "/util/doc", respectively. The files in the repository could be accessed from both OS/2 (the examples assumed that drive letter "u:" is mapped to "/util") and other UNIX systems over a network using the Network File System (NFS) capability in TCP/IP. A short OS/2 REXX command procedure, "9 CMDU.CMD", enabled executing UNIX commands from an OS/2 command line prompt.

Sometimes it is useful to execute a UNIX command with a specific user ID which does not identify a person. For example, to allow your user ID to read a file created by a process running under another user ID, the file access permissions can be changed using that ID. The file "CMDU.CMD" in Figure 1 is very similar to "CMDU.CMD" (Figure 7 in Part I), but uses "oracle" as the user ID. In this case, the REXX command file can be stored in the common repository directory "u:\cmd" and shared by users.

If "CMDU.CMD", has been added to the end of the "SET PATH" statement in CONFIG.SYS, an Oracle trace file's permissions can be displayed and changed by entering the following commands in an OS/2 Window (the examples assume \$ORACLE_HOME is on the "/u" file system):

```
cmdo ls -lt /u/oracle/rdbms/log/ora_*.trc
cmdo chmod 644 /u/oracle/rdbms/log/ora_#####.trc
```

The "#####" would be replaced by the UNIX process identifier (SPID) of one of the trace files displayed by the first command. This would allow someone without access to the "oracle" user ID to display the trace file, possibly using an OS/2 command like:

```
cmdu cat /u/oracle/rdbms/log/ora_#####.trc
```

The trace could easily be stored in a file or printed by adding ">myoracle.trc" or ">prn" to the end of the command.

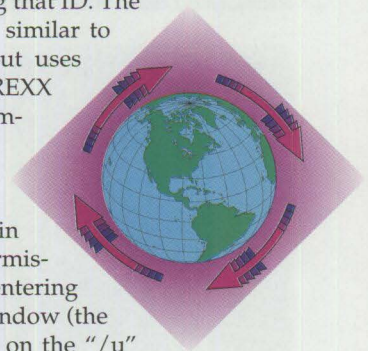


Figure 4: OS/2 REXX Command Procedure u:\cmd\o.cmd

```

/* O.CMD */
/* Re-displays output from last UNIX script */
parse arg script parms
if script = '' then do
  script = value("LAST_BIN_SCRIPT",,"OS2ENVIRONMENT")
  if script = '' then do
    say "Please specify the name of the script"
    exit
  end /* Do */
end /* Do */
tempdir = value("TMP",,"OS2ENVIRONMENT")
if tempdir \= '' then tempdir = tempdir || '\ '
file = tempdir||script".o"
if stream(file,'c','query exists') == '' then do
  file = 'u:\bin\script'
  if stream(file,'c','query exists') == '' then do
    say script "is not a valid UNIX script name"
    exit
  end /* Do */
else do
  say "There is no output from script" script
  exit
end /* Do */
end /* Do */
"@type" tempdir||script".o"
exit

```

Figure 5: UNIX C Shell Script /util/bin/lasttrc

```

ls -lt /u/oracle/rdbms/log/ora_*.trc | awk
+ 'NR==1{print "/util/bin/showtrc",$9,$6,$7,$8};NR>1{exit}
+ >temptrc
chmod 755 temptrc
temptrc
rm temptrc

```

Note:

+ indicates a line which is shown as a separate line but should be typed as a continuation of the previous line

Figure 6: UNIX C Shell Script /util/bin/showtrc

```

echo Trace file $1 was generated on $2 $3 at $4
echo
echo 'Line(s) of trace file identifying user:'
echo
grep ", user:" $1
echo
echo Last line of trace file:
echo
tail -1 $1

```

CREATING UNIX SHELL SCRIPTS

For more complicated commands or when it becomes necessary to execute more than one command, it may be helpful to create a UNIX shell script to execute the desired commands. For example, Figure 2 shows a script which can be used to set permissions on the Oracle trace files. The advantage is that the permissions for one or more files can be changed without having to specify the directory path for each one. If you tried to execute the "cd" and "chmod" commands separately with "cmdo", the effect of the change directory command would be lost before the second command was entered. To execute the "chmodtrc" script, type:

```
cmdo /util/bin/chmodtrc filename1 filename2 ...
```

at the OS/2 command prompt. The "chmodtrc" file must be made executable or you will get a message like "Permission denied." If you are not familiar with how to make shell scripts executable in UNIX, the procedure is illustrated in the sidebar on page 29. "Making UNIX Shell Scripts Executable." The full path "/util/bin/chmodtrc" is required because the ".login" or ".profile" file is not executed, so the path variable is not set.

Any UNIX command line utilities or applications can be used in your scripts. The primary objective is to provide a way to execute these scripts from OS/2 as well as UNIX.

The following command is used to establish what path is used for remote commands:

```
cmdo "env | grep PATH"
```

If the quotes were not included in this command, OS/2 would interpret the pipe character "|" rather than passing it through to UNIX and would treat "grep" as an OS/2 command receiving its input from the pipe. The OS/2 version of the command:

```
cmdo env | path "PATH"
```

produces the same output, but results in more data being transmitted from UNIX to OS/2 under the covers.

Another advantage of creating UNIX shell scripts is the commands can be executed from either OS/2 or UNIX. For example, the chmod command could be executed from the UNIX command prompt by typing:

```
chmodtrc ora_#####.trc ora_#####.trc ...
```

if you were already logged in as "oracle" and had "/util/bin" in the UNIX path. In this case, the PATH in the ".login" or ".profile" file in the home directory of the "oracle" user would need to be modified.

If we can execute these commands from UNIX, why did we bother with the effort to allow them to be executed from OS/2 as well? Here are a number of reasons:

- An OS/2 command prompt can be accessed more easily from the LaunchPad or Window List (I keep an OS/2 Window active most of the time) than a UNIX prompt can be by logging in through Telnet.

- Multiple systems can be accessed from a single OS/2 Window while individual Telnet windows would be needed to access multiple systems without logging in and out.

- The arrow keys can be used to recall and edit previous commands, making it easy to execute commands with minor changes or execute the same command on different systems by changing the "t" in "cmdt" (test) to a "p" (production), for example.

- The Pause key (or Ctrl+S) can be used to stop the output from scrolling off the screen. Any other key resumes scrolling.

- Output can be redirected to a file or printer.

- The Alarms applet of OS/2.1, the Launch option in the IBM Works Planner application in OS/2 Warp 3's BonusPak or

another OS/2 scheduling utility can be used to automatically execute scripts, instead of the UNIX "cron" utility.

OS/2 and TCP/IP for OS/2 are compatible with UNIX (except for the directory separator characters "/" and "\" in file path names), allowing this flexibility.

Let's say you want to learn how to write scripts for the UNIX C shell. Most of the information is in the "csh" manual page, which means you can display it with the UNIX command "man csh". But if you want to print it to a local printer or network printer assigned to "LPT1:", you could use the following OS/2 command:

```
cmdo man -pcat csh > prn
```

The parameter "-pcat" tells the "man" program to display the output using "cat", which simply sends all standard input to the standard output, instead of the default "pg", which would cause it to pause, waiting for keyboard input between pages.

ADDING "INTELLIGENCE"

This approach can be improved by taking advantage of some REXX capabilities and the fact that OS/2 can "see" the UNIX shell scripts in the repository. The file "BINO.CMD" in Figure 3 features a number of improvements:

- executes UNIX shell scripts in /util/bin;
- eliminates the need to specify full path names for scripts;
- verifies the existence of the script before trying to execute it;
- allows specifying a separate temporary directory;
- "remembers" the name of the last script executed;
- displays the name of the script being executed and its parameters; and
- saves the output from the script in a file in addition to displaying it.

The name of the temporary directory is specified by setting the "TMP" environment variable, which can be done automatically by adding a line such as "SET TMP=d:\TEMP" to CONFIG.SYS. You should create the directory on your hard disk with the OS/2 command "MKDIR d:\TEMP".

BINO.CMD also has a few limitations:

- It will not execute UNIX commands without a shell script.
- Multiple copies of the same script cannot be run simultaneously.
- Since the output is being redirected,

Making UNIX Shell Scripts Executable

1. Change to your home directory:

```
cd
```

2. Create a "bin" subdirectory under your home directory if you don't have one already. For example, to create "/usr/yourUsername/bin", use:

```
mkdir bin
```

3. Add your "bin" subdirectory to the path in your environment so that each script can be executed simply by typing its name. For example, if you are using the C shell, edit the ".login" script in your home directory:

```
vi .login
```

Find the following line:

```
set path = ($path .)
```

There may be additional subdirectories in the path, for example:

```
set path = ($path /u/oracle/bin .)
```

Add your "bin" subdirectory and the repository "bin" subdirectory somewhere in the path, like this:

```
set path = ($path $home/bin /util/bin /u/oracle/bin .)
```

Save the ".login" file. Log out and back in to set the path. Check to make sure the path was set correctly:

```
env | grep PATH
```

If you are using the Bourne or Korn shell, you would edit your ".profile" file in a similar manner, modifying the PATH by adding "/util/bin" and "\$HOME/bin", if it is not already there:

```
PATH=$PATH:$HOME/bin:/util/bin:/u/oracle/bin:/usr/bin
```

In this case, the PATH variable must be exported:

```
export PATH ...
```

If the ".profile" file had previously contained a PATH, the "export" command was probably there also.

4. Create the scripts. You can create them directly on the UNIX system or by using OS/2's System Editor (E) or Enhanced Editor (EPM) and move them to the UNIX system using NFS with OS2UNIX or FTP.

Most shell scripts on a UNIX system are written for the standard shell "sh". The C shell will execute a script under the standard shell unless the script starts with a comment. If you are using the C shell and include C shell commands such as "setenv" in your scripts, make sure the first line of the script starts with the comment character "#".

5. Make the script executable by changing the attributes of the script file:

```
chmod 755 yourscript
```

Since "csh" makes a list of files in the path when it is started, you must also relogin, start a new shell or execute the built-in shell command "rehash" if you are using the C shell.

You should now be able to execute the script just by typing its name at the UNIX shell prompt.

Figure 7: OS/2 REXX Command Procedure u:\cmd\binat.cmd

```

/*****
/* u:\cmd\binat.cmd
/*
/* Purpose:
/* Schedule UNIX script for later execution
/*
/* Usage:
/* binat hhmm date script parameters
/*
/* Examples:
/* binat 11:00 p.m. today script parameters
/* binat 0100 tomorrow script parameters
/* binat noon Saturday script parameters
*****/
parse arg time date script parms
if time = '' then
    time = '-1' /* hyphen, lowercase "L" */
else do
    file = 'u:\bin\'script
    if stream(file,'c','query exists') == '' then do
        say script "is not a valid UNIX script name"
        exit
    end /* Do */
end /* Do */
"@rexc yourHostname -l yourUsername -p yourPassword /bin/echo
+ 'csh /util/bin/"script parms"^|at" time date
exit

Note:
+ indicates a line which is shown as a separate line but
should be typed as a continuation of the previous line

```

Figure 8: Text File u:\doc\scripts.doc

```

REXX command procedures - u:\cmd
binat.cmd - Schedule a UNIX shell script for later execution:
    binat <time> <day> <script> [<parameters>]
bino.cmd - Execute a UNIX shell script with "oracle" user ID:
    bino <script> [<parameters>]
cmdo.cmd - Execute a UNIX command with "oracle" user ID:
    cmdo <command> [<parameters>]
man.cmd - Display documentation for a given script:
    man <script>
o.cmd - Re-display output from a UNIX shell script:
    o [<script>]

    If <script> is omitted, o.cmd displays the output from the
    last UNIX shell script executed.

UNIX scripts - /util/bin
To execute UNIX shell scripts from UNIX:
    <script> [<parameters>]
To execute UNIX shell scripts from OS/2, use bino.cmd:
    bino <script> [<parameters>]
chmodtrc - Change access permissions on Oracle trace files:
    [cmdo] chmodtrc ora_####.trc ora_####.trc ...
lasttrc - Display information from most recent Oracle trace file:
    [cmdo] lasttrc
showtrc - Display selected information from an Oracle trace file:
    [cmdo] showtrc <file> <month> <day> <time>

    showtrc is intended to be called from the lasttrc script
    rather than being executed directly

```

prompts will not be displayed until after the script completes. When the script requires input from the keyboard, it may seem to hang, but it will continue after the input has been provided. Even when developing scripts which receive their input

through parameters rather than the keyboard, this last point becomes a factor during testing. If a script hangs, it may be necessary to press Ctrl-C once or twice to cancel a script, then display the output to see what went wrong.

Another REXX command procedure can be created to re-display script output more easily. Figure 4 shows a procedure that will re-display the output from the last script executed in the current session simply by entering "O" on the OS/2 command line. To re-display the output from a particular script, include the name of the script after the "O" command, for example:

```
o chmodtrc
```

If the output file is not found a message is displayed to indicate whether no output exists or the script name is not valid.

USING UNIX UTILITIES

UNIX provides a variety of utilities which can be useful. The scripts in Figures 5 and 6 make use of the "awk", "grep" and "tail" utilities. This pair of scripts can be used to display information from the most recently generated Oracle trace file by entering the command:

```
bino lasttrc
```

at an OS/2 command prompt. Given a "program" and file as input, "awk" will execute the program for each line of the file, incrementing "NR", the number of records, for each line. In "/util/bin/lasttrc", "ls -lt ...ora_*.trc" lists the Oracle trace files in descending order and "NR==1" causes the print statement to be executed only for the first line of the file, the line with the most recent trace file. The variables "\$9,\$6,\$7,\$8" extract the file name, month, day and time fields and include them in a "showtrc" command which is built in a temporary script file named "temptrc". The awk program could also be placed in a file and referenced in the "awk" command like this:

```
ls -lt ... | awk -f lasttrc.awk > temptrc
```

There are many other ways UNIX utilities can be useful in an Oracle environment as well as other environments.

SUBMITTING SCRIPTS FOR LATER EXECUTION

Frequently it is necessary to submit a job to be executed at a later time. In a database environment, for example, it may be necessary to execute certain functions during times of least activity to avoid locking or performance problems. Figure 7 shows a REXX command which schedules a UNIX shell script at a specified time and date using the UNIX "at" command. To use this, enter an OS/2 command like this:

```
binat 2345 today rmarch 7
```

This command would execute the UNIX shell script "/util/bin/rmarch" at 11:45 p.m. Display the manual page for the UNIX "at" command to find the possible values for the time and date parameters. When no parameters are included, "binat" displays a list of currently scheduled commands. If the "rmarch" script contained the command:

```
find /u/arch -name arch\* -mtime +$1 -exec rm -f {} \;
```

it would remove all Oracle archive logs older than the number of days specified in the "binat" command. If "-mtime +3" was

added to the "find" command, you could ensure that archive logs less than four days old would not be deleted even if fewer days were specified in the "rmarch" command.

Note that for the "-mtime" and "-atime" parameters of the "find" command, the number of days since a file was modified or accessed is calculated first then truncated to an integer before it is compared with the number of days specified in the command. For example, if it is currently 11:30 a.m. on Friday, a file created at 11:45 a.m. on Monday is 3.99 days old which will be considered to be three days and will not be selected by a "find" command containing the parameter "-mtime +3".

Scripts could also be scheduled to execute at regular intervals using the UNIX "cron" utility. For example, the "rmarch" script could be scheduled to run at 5 a.m. daily. Create a file with the following line:

```
0 5 * * * csh /util/bin/rmarch 7
```

and save it as "mycrontab", for example. Login as "oracle" and execute the following command:

```
crontab mycrontab
```

to schedule the command. The "oracle" user ID needs to be included in the "/usr/lib/cron/cron.allow" file to allow execution of the "crontab" command. To schedule additional commands, add them to the existing "mycrontab" file, since the "crontab" command replaces the entire list.

The cron tables for different users could be stored in each user's home directory or you could add a "/util/cron" subdirectory to your repository to store the cron table files for all users.

DOCUMENTING THE SCRIPTS

It is a good idea to document the scripts as you create them. The more scripts you create, the more difficult it is to remember their names and what parameters are needed. As shown in Figure 8, a couple of lines for each script may be all that is needed. Using this format, help for a particular script can be extracted with the OS/2 "find" command or a REXX command procedure containing the "find" command such as the one in Figure 9.

STAY TUNED

This article presented an approach for executing UNIX shell scripts, providing examples which are applicable to an Oracle environment, even though it can be used for any UNIX function.

Part IV will present a number of scripts designed specifically for an Oracle database environment. The scripts will be written for the Oracle SQL*Plus utility and executed from either UNIX or OS/2 using a command procedure similar to "BINO.CMD". The concluding article, Part V, will present a simple C language program which cleans up the output and allows printed reports to be generated using the approach presented in the first four parts.

ADDENDUM: BEWARE

At 9:40 a.m. on the day I was to submit this article, after a weekend of extensive testing of an upgrade to the Oracle DBMS and SQL*Net on our primary production database server, we began experiencing a problem with SQL*Net SPX which prevented our 130 users from accessing the database. A group of three messages appeared repeatedly in the "\$ORACLE_HOME/spx/log/spxsrv.log" file:

```
ERROR: unable to read network string (9)
```

Figure 9: OS/2 REXX Command Procedure u:\cmd\man.cmd

```
/* MAN.CMD */
/* Display documentation in u:\doc\scripts.doc for given script */
parse arg script parms
cmdpath = 'u:\cmd\'
binpath = 'u:\bin\'
sqlpath = 'u:\dba\'
docpath = 'u:\doc\'
if script == '' then do
  say ""
  say "Usage:"
  say "  MAN <script>"
  say ""
  say "Parameters:"
  say "  <script> - name of script for which help is desired"
  exit
end /* Do */
file = cmdpath || script'.cmd'
if stream(file,'c','query exists') == '' then do
  file = binpath || script
  if stream(file,'c','query exists') == '' then do
    file = sqlpath || script'.sql'
    if stream(file,'c','query exists') == '' then do
      say script "is not a valid script name"
      exit
    end /* Do */
  end /* Do */
end /* Do */
'@echo off'
say ""
'find "'script'" <'docpath'scripts.doc'
exit
```

```
ORA-06771: TLI Driver: error reading version
TLI EVENT: connection indication received
```

Since the problem seemed to be affecting only the one system, we suspected the upgrade. However, neither backing off to the previous version of SQL*Net nor restarting the system as we had done before testing resolved the problem. We did notice that an SPX loopback in SQL*Plus:

```
sqlplus username/password@x:hostname:instance
```

would work only until a client on the network attempted to access the system, then would fail until the SQL*Net SPX process was killed and restarted. We also discovered that we could create the same problem with other database servers by crossing certain network bridges which keep groups of clients and servers isolated from the rest of the network.

After our group of five employees, two consultants and Oracle support had worked on the problem for 14 hours, it was finally tracked down to a system which unknown to us, had been brought up on the network that morning for the purpose of trying Windows NT for the first time.

Since our primary objective was to get the production application back up, it has not yet been determined whether this was a configuration problem or a bug in Windows NT.

Was this article of value to you? If so, please circle Reader Response Card No. 26.

NaSPA member Robert Simpson has more than 16 years computing experience, specializing in systems software support. He is experienced in installing and supporting OS/2 and related communications software, as well as data base and communications software on the MVS/ESA platform. He can be reached via CompuServe ID 71520,737 or Internet address 71520.737@compuserve.com.

Remote LAN Connectivity

BY JOHN E. JOHNSTON

One of the hottest trends in the industry today is remote computing. Employees demand access to corporate data from home, hotels or anywhere in between. Not only do these employees want access to the corporate data, but they also want to access every system in the enterprise, be it a mainframe, a mini-computer or a file server.

If you are a network administrator you have probably already been bombarded with requests to access the LAN remotely. I will discuss some of the options available today which will allow you to connect a remote OS/2 PC to your LAN.

REMOTE NODE, REMOTE CONTROL AND APPLICATION SERVERS

There are three prominent remote access methodologies in use today: remote node, remote control and the application server. Let's take a quick look at the functionality of each of these.

Remote node makes the remote workstation appear and act just like a locally-attached workstation. The remote node can access any service on the LAN just as a locally-attached node can. The remote node methodology is very robust as no special considerations must be made to the applications to support remote clients. The one noticeable drawback of remote node is speed, or should I say, lack thereof. The reason for this is obvious: this technology uses a 28,800 (or higher with compression) bits per second phone line to replace a 10 million (or 16 million for Token-Ring) bits per second LAN cable. This is much like using a drinking straw to drain a swimming pool. Emerging communications technologies, such as ISDN, should help this pipeline problem and make remote node a viable remote communications option.

Remote control allows a remote PC to take over a locally-attached LAN workstation. All of the processing occurs on the local PC and only the keystrokes and screen images are sent over the telephone wire. Remote control is much more responsive than remote node, but is not as robust. Remote control also requires the use of two PCs; one at the local site

and one at the remote site.

Application servers are relatively new on the scene and can provide a good compromise for remote connectivity. An application server slices up a PC and allows each remote user to run applications on one of the slices. This is much like remote control as all of the processing occurs at the local workstation and only the keystrokes and screen images are shipped over the wire.

We are using a 28,800 bits per second phone line to replace a 10 million bits per second LAN cable.

This is much like using a drinking straw to drain a swimming pool. Emerging communications technologies, such as ISDN, should help this pipeline problem and make remote node a viable remote communications option.

Some remote communications products also allow you to combine remote node and remote control giving you the best of both worlds. To utilize this hybrid you must first establish a remote node connection then utilize the local network to access the application server.

There are several OS/2 packages available that provide remote communication to a LAN. The space limitations of this column prohibit a discussion of every product so only a few will be presented.

Remote Node Using the IBM 8235

The IBM 8235 with its accompanying software allows OS/2, DOS or Windows clients to dial into a LAN as a remote node. (The 8235 is actually a re-marketed Shiva product.) The 8235 connects to a

Token-Ring or Ethernet LAN and includes serial ports for connecting modems. You will need the NetWare Requester installed on the remote PC in order to connect to a NetWare LAN via the 8235. You can also use Communications Manager/2 on the remote PC to connect to a mainframe or AS/400 using the 8235.

Remote Control Using OS2You

OS2You is a collection of shareware products which allow you to control an OS/2 system from a remote location. A variety of communications links are supported including dial-up modems, NetBIOS, Named Pipes, SPX and TCP/IP. OS2You is comprised of the following products:

■ **OS2You Remote Access Facility:** The base facility that enables you to remotely control OS/2 and DOS character mode programs.

■ **PM2You Remote Access Facility:** Does everything that OS2You does, but also enables you to remotely control OS/2 Presentation Manager (PM) programs.

■ **WinTerm Windows Client:** Terminal client that will allow you to remotely control OS/2 character mode and Presentation Manager sessions from Windows in combination with OS2You and PM2You.

■ **M2Zmodem File Transfer Option:** Allows you to perform file transfers between a host with OS2You/PM2You and a terminal.

■ **FAX2You Fax Receive Option:** Lets OS2You/PM2You accept both incoming data and fax calls with a faxmodem. When a fax call is detected the fax is received and printed.

You can obtain the OS2You suite of products in the OS2USER forum on CompuServe.

Application Server Using Citrix

The Citrix (Coral Springs, Fla.) system allows a remote OS/2 user to run applications on an application server attached to a network. There are many connectivity scenarios available with the Citrix

system. Citrix also allows you to combine remote node and remote control by supporting many of the remote node products, such as the IBM 8235. You can establish your remote node session via the 8235 then use the services of the network to access the Citrix application server.

LOOKING AHEAD

Remote connectivity will continue to be an important issue throughout the '90s. Remote node, being the most robust of the connectivity options, will continue to receive a great deal of attention from software and hardware vendors. As ISDN and data compression techniques mature we should see the bandwidth required for acceptable remote node operations become a reality. OS/2 has the opportunity to become a leader in remote connectivity. All of the aforementioned scenarios can be accomplished under DOS/Windows but after loading all of

the required drivers and TSRs the poor end user is left with very little conventional memory to work with. OS/2 can eliminate this completely. **ts**

If you have any questions, comments or ideas for future topics for this column, feel free to contact me via NaSCOM ID Johnjohe or CompuServe ID 73473,2146.

Was this column of value to you? If so, please circle Reader Response Card No. 38.



NaSPA member John E. Johnston is manager of technical support and communications for a major hospital in Pennsylvania. He designs and maintains cross-platform local and

wide area networks utilizing NetWare, OS/2, DOS and Windows.

Coming in May *technical* SUPPORT

Supporting Enterprise Networks and Operating Environments

- Managing Tables of Data in CICS
- High Performance Remote Access
- NetWare Statistics Boot Camp: Is Your Server Healthy?
- Multimedia Over Networks
- Managing the Clients in a NetWare and Windows Client/Server Environment
- UNIX, Oracle and OS/2 Coexistence

Learning How to Surf?

We have just the books to answer your questions about the Internet.

NaSPA is now offering two newly-released books from Prentice-Hall.

HANDS-ON INTERNET: A PC User's Guide

(book/disk package) by David Sachs and Henry Stair - \$29.95

A DOS USER'S GUIDE TO THE INTERNET: Email, Netnews and

File Transfer With UUCP (book/disk package) by James Gardner - \$34.95

To order, complete the following and either fax or mail it back to NaSPA/Library.

Name _____ USERID _____

Phone _____ Fax _____

Address _____

City _____ State _____

Country _____ Zip, Postal Code _____

Please send me:

☐ **HANDS-ON INTERNET: A PC User's Guide** by David Sachs and Henry Stair - \$29.95

☐ **A DOS USER'S GUIDE TO THE INTERNET: Email, Netnews and File Transfer With UUCP** by James Gardner - \$34.95

U.S. and Canadian orders add \$5.00 for shipping. International orders add \$18.00 (U.S. currency).

Please charge my credit card: ☐ VISA ☐ MasterCard

Card# _____ Exp. ____/____ Signature _____ Date _____

Return to: NaSPA/Library, 4811 S. 76th Street, Ste. 210, Milwaukee, WI 53220 Fax: 414-423-2433

